

Tinjauan Pustaka Sistematis terhadap Basis Data MongoDB

Renaldi¹, Billy Cahyo Santoso², Youzy Natasya³, Steven Willian⁴, Fladinand Alfando⁵

Teknik Informatika; Pradita University; Scientia Business Park, Jl. Gading Serpong Boulevard No.1, Curug Sangereng, Kelapa Dua, Banten 15810; (021) 55689999;

email : renaldi@pradita.ac.id¹, billy.cahyo@pradita.ac.id², youzy.natasya@pradita.ac.id³, steven.willian@pradita.ac.id⁴, fladinand.alfando@pradita.ac.id⁵

Abstrak: Istilah *NoSQL* di dunia *IT* sudah mulai terkenal di bidang basis data. Basis data *SQL* dan *NoSQL* memiliki perbedaan yang cukup signifikan. Model data yang digunakan basis data *SQL* yaitu berupa tabel, yang tersusun atas baris dan kolom. Basis data *SQL* manapun pasti *model data* yang sama dalam menyimpan arsip datanya, yaitu melalui tabel. Hal ini berbeda dengan basis data *NoSQL*. *No SQL* memiliki arti *non SQL* atau *not only SQL* yang artinya ditujukan pada penggunaan *model data* alternatif selain tabular (relasi antar tabel). Tergantung basis datanya, basis data *NoSQL* bisa berupa dokumen, grafik ataupun nilai kunci. Salah basis data *NoSQL* yang sudah banyak dikenal orang yaitu *MongoDB*. *MongoDB* adalah basis data *NoSQL* yang bersifat *document based*, artinya hanya tersusun atas koleksi dan dokumen. Paper ini tertulis tentang karakteristik *MongoDB*, Kelebihan dan kekurangan *MongoDB*, Aplikasi yang sudah menggunakan *MongoDB*, serta beberapa contoh query yang di gunakan di *MongoDB*. Dengan paper ini, penulis berharap pembaca bisa mengerti dengan benar apa itu *MongoDB* serta implementasinya.

Kata kunci: *MongoDB*, *NoSQL*, *Database*

Abstract: *NoSQL* is quite known method in databases making. *SQL* and *NoSQL* databases have some significant differences. *SQL* database contains tables and in tables, there are rows and columns. All kinds of *SQL* databases also use this kind of data model to save its data archive in tables. However, it is very different with *NoSQL* databasess. *NoSQL* stands for non *SQL* or not only *SQL* which indicates the use of alternative data archiving besides the traditional tabular method (relations between tables). *NoSQL* database is able to save documents, graphical design, and even key value, depending on the database used. One example of the widely used *NoSQL* database is *MongoDB*. *MongoDB* is a document-based *NoSQL* database, which means, only composed of collections and documents. This paper writing writes about the characteristics, advantages and disadvantages of *MongoDB*, applications that use *MongoDB* as the database, and also the examples of the query used in *MongoDB*. Through this writings, we hope that our readers could understand fully about what is *MongoDB* along with its implementation.

Keywords: *MongoDB*, *NoSQL*

1. Pendahuluan

Perkembangan *Big Data* di Indonesia terus meningkat dan hampir menyentuh segala aspek kehidupan. Mulai dari perindustrian, kesehatan, *E-commerce*, media sosial, bahkan pemerintahan. Dengan munculnya istilah *Big Data* ini maka diperlukan model data yang memiliki skema yang fleksibel dimana basis data relasional yang model datanya berupa tabel dan memiliki keterbatasan dalam menangani jumlah data yang besar dan beragam. Untuk itu diperlukan kelas basis data dan model data yang baru, sehingga istilah “*NoSQL*” diciptakan. Model data *NoSQL* tidak berupa tabel yang tersusun atas baris dan kolom. Namun, basis data *NoSQL* bisa berupa dokumen, grafik atau nilai kunci (Fathansyah, 2018). Seiring perkembangan zaman sudah banyak basis data *NoSQL* yang populer, salah satunya adalah *MongoDB*. *MongoDB* adalah basis data *NoSQL* yang bersifat *document based*. *MongoDB* bersifat document based artinya *MongoDB* tidak memiliki tabel, kolom ataupun baris. *MongoDB* hanya memiliki koleksi dan dokumen. Data yang disimpan dalam basis data *MongoDB* berupa file *JSON* yang disebut dengan istilah *BSON* (*Binary JSON*). Sistem basis data *MongoDB* menggunakan *key-value*, artinya setiap dokumen dalam *MongoDB* dipastikan memiliki *key* (Putra & Rahmayeni, 2016).

Karena terdapat fitur-fitur yang berbeda dengan basis data *SQL*, maka diperlukan pemahaman lebih lanjut. Paper ini dibuat dengan tujuan pembaca bisa paham mengenai fitur-fitur yang dimiliki *MongoDB*, Kelebihan dan kekurangan *MongoDB*, Aplikasi *MongoDB*, serta contoh penggunaan *query* pada *MongoDB*

2. Metode Penelitian

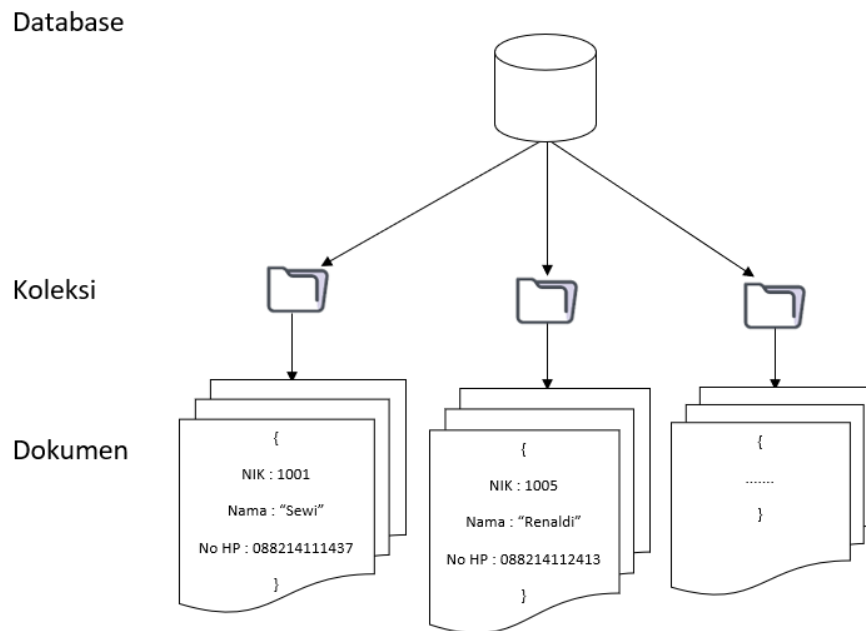
Metode penelitian dilakukan dengan metode *Systematic Literature Review* yaitu dengan mengidentifikasi, mengevaluasi dan menginterpretasi penelitian-penelitian yang sudah ada sebelumnya seputar basis data *MongoDB* untuk menjawab pertanyaan-pertanyaan :

1. Karakteristik basis data *MongoDB* termasuk perbedaannya dengan basis data relasional
2. Kelebihan dan kekurangan basis data *MongoDB*
3. Implementasi dan aplikasi yang sudah menggunakan *MongoDB*
4. Beberapa contoh penggunaan *query* pada basis data *MongoDB*

3. Hasil dan Pembahasan

3.1 Karakteristik basis data *MongoDB* termasuk perbedaannya dengan basis data relasional

Didalam database relasional data disimpan dalam satu entitas yang dapat saling berhubungan. Perhubungan tersebut disebut relasi. didalam entitas terdiri dari beberapa atribut yang akan menjadi nama kolom didalam sebuah tabel yang akan dikaitkan dengan domain tertentu seperti integer atau string. Sedangkan *MongoDB* data disimpan dengan berorientasi sebuah dokumen yang memiliki format JSON atau BSON. Dokumen juga mendukung penggunaan seperti lists, pointers, embedded arrays atau nested document sehingga menyederhanakan akses data.



Gambar 1. Struktur Database *MongoDB*(Sumber: *ANALYSIS AND COMPARISON OF DOCUMENT-BASED DATABASES WITH SQL RELATIONAL DATABASES: MONGODB VS MYSQL*)

Perbedaan lainnya ada pada sistem transaksional. Didalam database relasional menggunakan model ACID(Atomicity, Consistency, Isolation, dan Durability) yang gunanya untuk menjaga konsistensi data dan keteraturan data. Sedangkan didalam *MongoDB* menggunakan BASE (Basically available, Soft state, with Eventual consistency) (Deari, Zenuni, Ajdari, Ismaili, & Raufi, 2018).

Dengan dipertimbangkannya ukuran data penggunaan memori dan efisiensi bandwidth ketika dilakukannya proses transmisi data maka dalam MongoDB maksimal penyimpanan data hanya sampai 16 MB. Limitasi penyimpanan dapat diatasi dengan GridFS yang memecah data menjadi beberapa data berukuran kecil. MongoDB memiliki berbagai macam fitur, seperti *Map reduce based Aggregation Framework*, hampir mirip dengan Group By didalam MySQL, dalam MongoDB digunakan untuk agregasi, umumnya digunakan untuk memproses data dalam volume besar secara paralel. Lalu ada *Schema Less Database* dengan fitur ini database tidak mempunyai struktur data yang tetap. *Replication and fail-over support* dengan fitur ini MongoDB mendukung adanya replikasi data dengan menggunakan replica set. Dalam replica set, satu node didefinisikan sebagai node primer sementara semua node lainnya diklasifikasikan sebagai sekunder. Semua operasi tulis ditugaskan ke simpul primer, simpul sekunder dapat melakukan operasi baca (Winaya & Ashari, 2016).

3.2 Kelebihan dan kekurangan basis data *MongoDB*

Bagian ini kami bagi menjadi 2 bagian, yaitu kelebihan dan kekurangan dibandingkan dengan database SQL, dan yang satu lagi dibandingkan dengan database NoSQL. Untuk perbandingan dengan database SQL, kami menggunakan database MySQL sebagai bahan perbandingan. Sedangkan untuk database NoSQL kami membandingkannya dengan Cassandra.

3.2.1. Perbandingan MongoDB dengan database SQL

Pertama adalah perbandingan dengan database SQL, bagaimana kinerja kedua database dalam aplikasi file multimedia berbasis web. Kedua database akan diletakkan dalam sebuah server Linux dengan konfigurasi minimum, yang dibantu oleh server web Apache dan interpreter PHP. Semua file yang tersimpan akan dapat diakses dengan memakai aplikasi multimedia berbasis web. Data yang akan disimpan pada basis data kedua database ini berupa atribut atau metadata dari file multimedia pada umumnya yaitu judul, artis, kategori, jenis file, dan ukuran file – dan data binary dari file multimedia itu sendiri.

Beberapa keunggulan dan kekurangan dari database MongoDB dibandingkan MySQL adalah :

- a. Penggunaan processor (CPU) dan memori untuk MongoDB lebih sedikit daripada MySQL. MongoDB lebih cepat sekitar 3,6 kali lebih cepat daripada MySQL dalam penyimpanan file berukuran kecil. Dari 2 hal tersebut bisa disimpulkan bahwa MongoDB lebih unggul dalam kecepatan serta keiritan dalam hal pemakaian sumber daya komputer dibandingkan MySQL untuk operasi Insert atau Create data binary. Operasi penghapusan file berukuran besar juga terlihat lebih efisien pada database yang menggunakan MongoDB.
- b. Dalam penghapusan file berukuran kecil MySQL lebih unggul dibanding MongoDB meski tidak ada perbedaan yang signifikan. Dalam menyimpan serta mengunduh file berukuran besar, penggunaan memori MongoDB lebih besar dibandingkan MySQL, sehingga dapat dikatakan MySQL lebih optimal dibandingkan MongoDB.

(Silalahi & Wahyudi, 2018)

Tabel 1 Perbedaan performa antara proses insert MySQL dan MongoDB (Sumber : PERBANDINGAN PERFORMANSI DATABASE MONGODB DAN MYSQL DALAM APLIKASI FILE MULTIMEDIA BERBASIS WEB)

Data	Processor (usr)		Memory (MB) (used)		Swap (MB) (used)		Time (s)	
	MySQL	MongoDB	MySQL	MongoDB	MySQL	MongoDB	MySQL	MongoDB
1	7	7	86	19	1165	409	3.476743	1.2549
2	15	8	21	2	0	408	0.877605	0.218272
3	16	7	36	18	0	408	0.921814	0.331327
4	14	11	39	4	0.128906	388	1.165534	0.232279
5	15	9	21	6	0.089844	392	0.943982	0.313019
6	12	7	70	26	0.085938	392	1.544793	0.201332
7	10	5	59	357	0.085938	390	4.357289	0.870136
8	16	10	48	12	0.085938	389	1.17692	0.377571
9	14	16	39	13	0.085938	388	1.265624	0.213503
10	20	11	52	14	0.085938	388	1.244093	0.249795
11	16	14	42	11	0.085938	371	1.477015	0.316518
12	16	11	68	24	7.136719	374	8.557555	0.719089
13	16	8	85	12	8.214844	391	4.214284	1.665814
14	21	7	75	19	9.253906	392	4.387183	1.045734
15	11	19	87	21	14	394	9.4808	2.274673
16	15	17	70	10	17	396	4.526862	1.300158
17	14	7	77	20	23	403	7.510885	2.544604
18	15	11	82	19	28	412	9.601188	1.819042
19	15	50	88	41	36	419	13.03361	3.572581
20	19	23	91	18	37	426	6.498529	3.571212
	14.85	12.9	61.8	33.3	67.26699	396.5	4.313115	1.154578

Tabel 2 Perbedaan performa antara proses delete MySQL dan MongoDB (Sumber : PERBANDINGAN PERFORMANSI DATABASE MONGODB DAN MYSQL DALAM APLIKASI FILE MULTIMEDIA BERBASIS WEB)

Data	Processor (usr)		Memory (MB) (used)		Swap (MB) (used)		Time (s)	
	MySQL	MongoDB	MySQL	MongoDB	MySQL	MongoDB	MySQL	MongoDB
1	9	11	10	7	298	307	9.195436	2.902609
2	12	10	12	3	300	307	10.82128	1.960059
3	12	11	15	3	302	307	13.45522	2.063635
4	12	8	11	5	302	307	21.54709	2.888499
5	10	11	20	6	307	307	20.92196	3.177332
	11	10.2	13.6	4.8	301.8	307	15.1882	2.598427

Tabel 3 Perbedaan performa antara proses read MySQL dan MongoDB (Sumber : PERBANDINGAN PERFORMANSI DATABASE MONGODB DAN MYSQL DALAM APLIKASI FILE MULTIMEDIA BERBASIS WEB)

Data	Processor (usr)		Memory (MB) (used)		Swap (MB) (used)	
	MySQL	MongoDB	MySQL	MongoDB	MySQL	MongoDB
K	4	3	225	140	0	0
S	6	6	385	749	337	0
B	26	5	522	82	331	0

3.2.2. Perbandingan MongoDB dengan database NoSQL lain

Bagian kedua ini membandingkan database NoSQL dengan sesama database NoSQL, bagaimana kinerja kedua database sebagai backend IoT Data Storage. Kinerja database dilihat melalui penggunaan CPU dan disk I/O, penggunaan memori, throughput serta runtime dalam melakukan operasi insert. Data yang dipakai pada pengujian berupa data string dan data gambar dengan variasi jumlah data antara 10.000, 30.000, 50.000, 70.000, 100.000, 120.000, 150.000 untuk data string serta 1000, 3000, 5000, 7000, 10000 untuk data gambar

Beberapa keunggulan dan kekurangan dari database MongoDB dibandingkan Cassandra adalah :

- Kinerja saat insert data string, dapat dilihat bahwa MongoDB lebih efisien dari segi runtime, throughput, penggunaan CPU serta disk I/O. Pada saat insert data gambar, MongoDB juga lebih efisien dari segi penggunaan memori.
- Cassandra memiliki kinerja yang lebih baik saat melakukan insert data gambar. Database ini lebih efisien dalam hal insert data gambar, runtime, throughput, serta nilai disk I/O yang lebih baik. Cassandra juga lebih baik dalam efisiensi penggunaan memori saat insert data string.

(Kurniawan, Pramukantoro, & Trisnawan, 2019)

3.2.3. Kelebihan dan kekurangan MongoDB secara umum

a. Kelebihan :

MongoDB memiliki format yang cepat karena mampu mencached dan model data berbasis dokumen, berbentuk seperti file JSON yang disebut BSON. Model data nya yang berbasis dokumen membuat penggunaanya untuk tidak perlu merancang struktur tabel seperti pada SQL. MongoDB bisa membuat struktur tabelnya sendiri secara otomatis pada saat melakukan Insert. Karena ini skemanya lebih fleksibel. Sebagai basis data NoSQL, MongoDB memiliki penyimpanan data yang lebih besar serta *low-cost*.

b. Kekurangan :

Belum banyak hosting web yang mensupport MongoDB. Tiap basis data NoSQL memiliki model data yang berbeda-beda, sehingga ketika berpindah dari basis data NoSQL satu ke NoSQL lain, perlu belajar dari awal. Berbeda dengan database-database SQL yang memiliki model data yang sama yaitu berbentuk tabel sehingga ketika berpindah dari database SQL satu ke database SQL lain tidak perlu belajar dari awal karena setidaknya memiliki kemiripan

(Farozzi, 2019) (Junaidi, 2016)

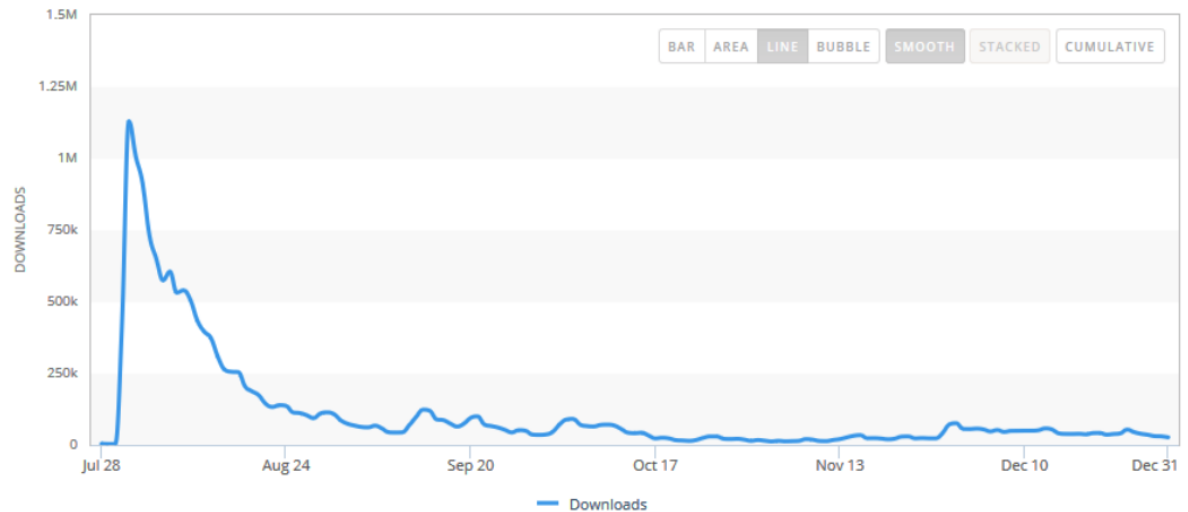
3.3 Implementasi dan aplikasi yang sudah menggunakan *MongoDB*

MongoDB sebagai basis data *NoSQL* yang sudah cukup terkenal, banyak perusahaan-perusahaan besar yang sudah menggunakannya seperti : SEGA, Coinbase, Medtronic, EA, Google, Adobe, facebook, PayPal, Nokia, Telefonica, Pearson, eBay dan lain sebagainya. Bukan hanya perusahaan besar, namun *MongoDB* juga diimplementasikan dalam *smart city* di Kota Chicago, Amerika Serikat (mongoDB, t.thn.)

Berikut penggunaan *MongoDB* oleh beberapa perusahaan yang tadi disebutkan namanya diatas :

3.3.1. Penggunaan *MongoDB Atlas* oleh SEGA HARDlight dalam *game* buatannya.

SEGA adalah salah satu perusahaan game ternama di dunia, terkenal dengan game iconiknya, Sonic the Hedgehog. SEGA memiliki banyak studio yang bergerak di bidang-bidang *Game Development* salah satunya adalah studio SEGA HARDlight yang bergerak di bidang *Mobile Game Development*.



Berbeda dengan *PC gaming* ataupun *Console gaming*, *Mobile gaming* lebih dinamis. *Mobile gaming* membutuhkan fleksibilitas dan kreativitas yang bisa berubah-ubah dengan cepat.

Gambar 2 *Spiky traffic for a popular mobile game, Crazy Taxi: City Rush*
(Sumber: <https://www.mongodb.com/blog/post/sega-hardlight-migrates-to-mongodb-atlas-simplify-ops-improve-experience-mobile-gamers>)

Mobile game memiliki pola grafik yang unik. Ketika sebuah mobile game pertama kali di rilis , jumlah pemain game akan memuncak, namun setelah beberapa hari bahkan jam, jumlah pemain akan menurun secara drastis. Yordan Gyurchev, *Technical Director* dari SEGA HARDlight mengatakan bahwa dengan volume

perubahan jumlah pemain sebesar ini, bahkan sedikit perubahan dalam *database* dapat memberi dampak yang besar. Dari masalah ini sudah dipastikan bahwa basis data *SQL* tidak efektif dalam mengatasi masalah tersebut, sehingga dibutuhkan alternatif lain yaitu penggunaan basis data *NoSQL*.

SEGA HARDlight memilih *MongoDB Atlas*, *cloud* yang di *host* layanan *MongoDB*. Alasan SEGA HARDlight menggunakan *MongoDB Atlas* sebagai basis data *NoSQL* nya adalah karena *MongoDB* jauh lebih mudah digunakan daripada *Cassandra* dan *Couchbase*. *Cassandra* dan *Couchbase* dinilai lebih kompleks. *MongoDB* memiliki skalabilitas dan fleksibilitas dalam mengatasi volume data yang besar juga menjadi salah satu alasan kenapa SEGA HARDlight memilihnya sebagai basis datanya.

Sekarang SEGA HARDlight sudah mengimplementasikan *MongoDB Atlas* di game-game buatannya seperti *Kingdom Conquest : Dark Empire* dan *Sonic Forces : Speed Battle*. Yordan mengatakan bahwa dengan menggunakan *MySQL* membutuhkan lebih banyak percobaan untuk mendapatkan *Backend* yang tepat untuk gamenya, Namun dengan *MongoDB* hanya perlu satu kali *launch*. (MongoDB/blog, t.thn.)

3.3.2. Medtronic, melakukan pengelolaan Data menggunakan MongoDB

Medtronic adalah perusahaan penyedia peralatan medis di Amerika Serikat. Medtronic yang bergerak dibidang medis sudah dipastikan mendapatkan sangat banyak *sample data* untuk peralatan medisnya setiap hari, yang didapatkan dari pasien-pasien yang menggunakan peralatannya. Karena hal ini, Medtronic tentunya menyimpan banyak sekali data yang beragam.

Untuk mengelola data yang kompleks tersebut, Medtronic menggunakan *MongoDB* sebagai basis data nya. Data yang dikelola Medtronic adalah data yang dinamis, sering berubah-ubah. *RDBMS* memiliki kelemahan dalam mengelola data yang seperti itu, oleh karena itu Medtronic pindah dari penggunaan *RDBSM* ke salah satu basis data *NoSQL* yaitu *MongoDB*.

Medtronic merasa bahwa *MongoDB* memiliki model data yang terbilang sederhana. Melakukan analisis setiap komponen bisa dilakukan dengan *query* yang cepat. Tidak perlu melakukan *Join* seperti pada *RDBSM*. (MongoDB/blog, t.thn.)

3.3.3. EA dalam game EA FIFA Online 3

EA adalah developer game yang fokus pada pengembangan game-game bertema olahraga. Salah satu gamenya yang terkenal di dunia adalah game FIFA. Seiringnya waktu, game-game buatannya bisa dimainkan secara online, seperti EA FIFA Online 3.

EA FIFA Online 3 adalah game online yang memiliki kebutuhan untuk mengskala jutaan pemainnya. EA FIFA Online 3 dikembangkan dalam *MongoDB* sebagai basis datanya. Alasannya karena *MongoDB* memiliki fitur-fitur yang membuatnya mudah untuk diskala. (MongoDB/blog, t.thn.)

3.3.4. Implementasi MongoDB dalam Adobe Experience Manager

Adobe membangun *persistance layer* untuk Experience Manager versi 6.0. menggunakan basis data *mongoDB*. Didalam *persistance layer* tersebut tersimpan banyak sekali data pelanggan hingga mencapai beberapa *petabyte*. Data pelanggan tersebut digunakan untuk konten *repository* berskala besar serta mengelola dan memberi *digital experience* yang baik.

MongoDB merupakan basis data yang memiliki solusi dalam skalabilitas dan aksesibilitas untuk permintaan bisnis sekarang. Skalabilitas dan aksesibilitas dari *MongoDB* mampu mengelola *repository* besar dari aset-aset digital yang memiliki ledakan perkembangan yang pesat.
(MongoDB/blog, t.thn.)

3.3.5. Implementasi MongoDB oleh Coinbase untuk membangun sistem yang terpercaya dalam menghadapi *Crypto Mania*

MongoDB memiliki model data yang fleksibel. Setiap kali sebuah produk baru ditambahkan, *MongoDB* dengan model data berbasis dokumen (*Document-based model data*) bisa membantu Coinbase untuk memperbarui data struktur yang ada yang dibutuhkan dengan cepat.

Coinbase menggunakan *MongoDB* dalam beberapa hal seperti:

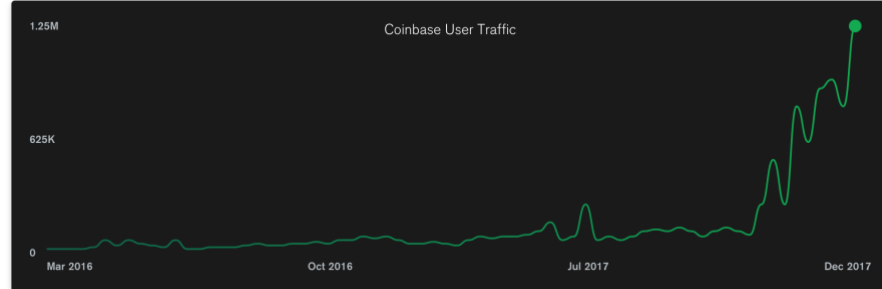
- System of Engagement* dimana tindakan *user* dan *event* terjadi pada saat *user session*
- System of Record* untuk menyimpan identitas *user*, informasi akun, transaksi, metode pembayaran dan lain sebagainya.
- System of Insight* untuk mendukung dan meningkatkan *Customer experience*



Gambar 3 : penggunaan *MongoDB* dalam sistem Coinbase(Sumber : <https://www.mongodb.com/customers/coinbase>)

MongoDB mempercepat perkembangan Coinbase karena menghilangkan kesalahan *Object-relational* yang biasanya disebabkan karena penggunaan basis data dengan model tabel seperti SQL.

MongoDB juga membantu Coinbase dalam menghadapi ledakan ketertarikan publik terhadap *digital assets* karena harga *Ethereum* yang melonjak tinggi pada pertengahan 2017.



Gambar 4 : Coinbase User Traffic (Sumber : <https://www.mongodb.com/customers/coinbase>)

Kapitalisasi pasar dari *Crypto currency* tertinggi dari Desember 2017 ke Januari 2018 dalam Coinbase menjadi aplikasi nomor satu pada saat itu. Coinbase menghadapi 1.2 juta RPM (*Request per Minute*) tapi kerja tim yang sudah dilakukan termasuk penggunaan *MongoDB* dalam mengimplementasikan *monitoring*, maka tidak ada masalah yang dihadapi.
(MongoDB/customer, t.thn.)

3.3.6. Implementasi MongoDB dalam Chicago Smart City

Dalam pengembangan smart citynya, Kota Chicago mengembangkan sebuah sistem bernama *WindyGrid* yang dibangun dengan platform *MongoDB*. *WindyGrid* mampu menghubungkan tujuh juta data di kota setiap hari. *WindyGrid* menggabungkan *MongoDB* dengan *Visual map* sehingga mampu mengelola kota secara *real time*.

WindyGrid dengan kombinasi data, peta serta analisis pada titik korelasi, mampu mengidentifikasi potensi masalah sebelum masalah tersebut terjadi. Ini bisa karena bantuan *MongoDB*. *MongoDB* memiliki model data yang fleksibel, ketika ada data baru yang muncul, *WindyGrid* tidak perlu kembali ke titik awal dan mengerjakan kembali skema datanya. Skema data tersebut bisa berevolusi sendiri secara *real time*. Ini adalah poin utama karena *WindyGrid* dipastikan mendapatkan banyak data terstruktur dan data tidak terstruktur setiap harinya. *MongoDB* bisa menyimpan data apapun dan bisa menganalisisnya dengan fitur *indexing*.
(mongoDB/customer, t.thn.)

3.4 Beberapa contoh penggunaan query pada basis data MongoDB

3.4.1.

`db.nama_table.find({query,projection})`

Query ini digunakan untuk menampilkan semua data yang ada pada Collection. Query ini sama fungsinya dengan query SQL.

`SELECT * FROM`

Kalau di dalam kurung kurawal terisi, maka, fungsinya sama dengan query SQL dibawah ini :

```
SELECT * FROM collection WHERE query,projection
```

3.4.2.

```
db.nama_table.updateOne(  
  { item: "field" },  
  {  
    $set: { "query, projection" },  
    $currentDate: { lastModified: true }  
  }  
)
```

Query MongoDB ini digunakan untuk mengupdate record di suatu field. Untuk query sqlnya, mempunyai query seperti :

```
UPDATE nama_table SET column1 = value1, column2 = value2, ... WHERE condition;
```

3.4.3.

```
db.inventory.deleteMany({parameter})
```

Query MongoDB ini digunakan untuk menghapus semua record yang cocok dengan parameter di satu collection.

Untuk mendelete satu data yang cocok dengan parameter di satu collection, digunakan query
`db.inventory.deleteOne({parameter})`

3.4.4.

use database_name

Query MongoDB ini digunakan untuk membuat database di MongoDB dan akan membuat database jika belum ada database yang bernama itu.

Jika ingin membuat satu collection, maka kita bisa memakai query ini

```
db.createCollection(name, options)
```

(Bradshaw, Brazil, & Kristina, 2019)

(Docs MongoDB, t.thn.)

4. Kesimpulan

Dari *study literature review* yang kami lakukan, diketahui bahwa *MongoDB* merupakan basis data *NoSQL* berbasis dokumen (*document based*). *MongoDB* menyimpan datanya dalam bentuk file seperti *JSON* yang disebut *BSON (Binary JSON)*. *MongoDB* memiliki fitur-fitur seperti lists, pointers, embedded arrays atau nested document sehingga menyederhanakan akses data, *Map reduce based Aggregation Framework* untuk agregasi, *Schema Less Database* untuk struktur data yang fleksibel, *Replication and fail-over support* untuk mendukung replikasi data. Dari data yang didapatkan, bisa dikatakan bahwa *MongoDB* memiliki performansi yang baik dibandingkan dengan *MySQL*.

Dari perusahaan-perusahaan yang memilih *MongoDB* sebagai basis datanya, bisa disimpulkan alasan kenapa mereka memilih *MongoDB* :

- MongoDB* memilki fleksibilitas yang tinggi karena memilki fitur *Schema Less Database*
- Memilki perfoma yang bagus
- High availability*
- Memilki banyak fitur meningkatkan skalabilitas dan fleksibilitas
- Mudah dalam penggunaan, penyetelan dan di *maintain*.

- f. Mampu mengolah data dalam jumlah yang sangat besar dan data yang tidak tetap.

Daftar Referensi

- Bradshaw, S., Brazil, E., & Kristina, C. (2019). MongoDB: The Definitive Guide: Powerful and Scalable Data Storage. Dalam S. Bradshaw, E. Brazil, & C. Kristina, *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage* (hal. 514). Sebastopol: O'Reilly Media, Inc.
- Deari, R., Zenuni, X., Ajdari, J., Ismaili, F., & Raufi, B. (2018). ANALYSIS AND COMPARISON OF DOCUMENT-BASED DATABASES WITH SQL RELATIONAL DATABASES: MONGODB VS MYSQL. *Proceedings of the International Conference on Information Technologies*, 1-10.
- Docs MongoDB. (t.thn.). Diambil kembali dari MongoDB: <https://docs.mongodb.com/manual/tutorial/query-documents/>
- Farozi, M. (2019). DESAIN BASIS DATA NON RELASIONAL NOSQLMONGODB PADA WEBSITE SISTEM INFORMASI AKADEMIK. *Jurnal Sistem Informasi Komputer dan Teknologi Informasi (SISKOMTI)*, 24-39.
- Fathansyah. (2018). *BASIS DATA Revisi Ketiga*. Bandung: INFORMATIKA.
- Junaidi, A. (2016). Studi Perbandingan Performansi Antara MongoDB. *ANNUAL RESEARCH SEMINAR 2016*, 460-465.
- Kurniawan, A. K., Pramukantoro, E. S., & Trisnawan, P. H. (2019). Perbandingan Kinerja Cassandra dan MongoDB Sebagai Backend IoT Data Storage. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 364-371.
- mongoDB. (t.thn.). Diambil kembali dari mongoDB website: <https://www.mongodb.com/who-uses-mongodb>
- MongoDB/blog. (t.thn.). Diambil kembali dari MongoDB: <https://www.mongodb.com/blog/post/sega-hardlight-migrates-to-mongodb-atlas-simplify-ops-improve-experience-mobile-gamers>
- MongoDB/blog. (t.thn.). Diambil kembali dari MongoDB: <https://www.mongodb.com/blog/post/how-medtronic-manages-machine-data-mongodb?c=d67bc669bb>
- MongoDB/blog. (t.thn.). Diambil kembali dari MongoDB: <https://www.mongodb.com/blog/post/ea-scores-mongodb-based-fifa-online-3?c=1534ec3d9c>
- MongoDB/blog. (t.thn.). Diambil kembali dari MongoDB: <https://www.mongodb.com/press/mongodb-delivers-multi-petabyte-data-store-option-adobe-experience-manager?c=69bb399984>
- mongoDB/customer. (t.thn.). Diambil kembali dari mongoDB: <https://www.mongodb.com/customers/city-of-chicago>
- MongoDB/customer. (t.thn.). Diambil kembali dari MongoDB: <https://www.mongodb.com/customers/coinbase>
- Putra, E. K., & Rahmayeni, F. (2016). IMPLEMENTASI DATABASE MONGODB UNTUK SISTEM INFORMASI BIMBINGAN KONSELING BERBASIS WEB. *TEKNOIF*, 67-73.
- Silalahi, M., & Wahyudi, D. (2018). PERBANDINGAN PERFORMANSI DATABASE MONGODB DAN MYSQL DALAM APLIKASI FILE MULTIMEDIA BERBASIS WEB. *Computer Based Information System Journal*, 65-77.
- Winaya, I. G., & Ashari, A. (2016). Transformasi Skema Basis Data Relasional Menjadi Model Data Berorientasi Dokumen pada MongoDB. *IJCCS*, 47-58.